BRIEF ยอดเอดิเตอร์

สำหรับ

โปรแกรมเมอร์

เอนก ศิรจริยาภิวัฒน์

ความเป็นมา

คำว่า "เอดิเตอร์" นั้นหลาย ๆ คน คงจะรู้จักและได้นำมาใช้บ้างแล้ว ตั้งแต่ยุคโปรแกรมจัดระบบงานบนเครื่องแอป-เปิ้ลทู คือซีพีเอ็ม (CP/M) เราก็ใช้ ED ช่วยในการเขียนโปรแกรม ต่อจากนั้นที่เรารู้จักกันดีก็คือ เอดิเตอร์ในเทอร์โบปาสคาล และเมื่อมาถึงยุคของเครื่อง

ไมโครฯไอบีเอ็มพีซี เราก็ได้มีโอกาสใช้โปรแกรมไซด์คิก (Sidekick) ม้ารับใช้ตัวเก่ง ซึ่งเป็นโปรแกรมเอดิเตอร์ที่นิยมใช้กันแพร่หลายมาก และล่าสุดก็ไซด์คิกพลัส (Sidekick Plus) ผู้ช่วยคนใหม่ที่มีการเพิ่มฟังก์ชันการใช้งานต่าง ๆ ให้มีความสะดวกสบายสำหรับผู้ใช้มากขึ้น เมื่อก่อนเราใช้ ED หรือ EDLIN มันก็ไม่สะดวกสำหรับเรานัก แต่เมื่อมาถึงยุคของเอดิเตอร์ที่มีการทำงานแบบฟูลสกรีนเอดิเตอร์ (full screen editor) หลาย ๆ อย่างก็สะดวกสบายขึ้นมาก แต่ท่านเคยเจอเหตุการณ์เหล่านี้กับเอดิเตอร์ของท่านหรือไม่ เช่น ต้องการลบโปรแกรมเพียงไม่กี่บรรทัด แต่ด้วยความมันมือ ท่านก็กด CRTL-Y จนเพลิน ทำให้เลยบรรทัดที่ต้องการลบเสียแล้ว เพราะคีย์ค้างในบัฟเฟอร์ (Buffer) การลบบล็อกผิด หรือ ปัญหาในเรื่องการดูข้อความบางอย่างที่ลบไปแล้ว...ฯลฯ เมื่อปัญหาเหล่านี้เกิดขึ้นจะต้องทำอย่างไร ท่านคงอยากได้ฟังก์ชันพิเศษที่ลบข้อความไปแล้วสามารถเรียกข้อความที่ลบไปแล้วกลับคืนมาได้ นั่นก็คือ ฟังก์ชัน UNDO ซึ่งเอดิเตอร์ที่ผมจะแนะนำนั้นจะมีฟังก์ชัน UNDO นี้ด้วย และอีกหลาย ๆ ฟังก์ชันที่น่าสนใจ เอดิเตอร์ที่กล่าวถึงนี้ก็คือ บรีฟ (Brief) ข้อมูลทางเทคนิคของบรีฟนั้นดังรูปที่ 1

ผู้เขียน	David B. Nanian และ Michael Strickman
บริษัท	Under Ware Inc.
ดอส	เวอร์ชัน 2.0 หรือ สูงกว่า
จำนวนบัฟเฟอร์	ไม่มีขีดจำกัด ขึ้นอยู่กับหน่วยความจำที่มีอยู่
ขนาดบัฟเฟอร์	ไม่มีขีดจำกัด ขึ้นอยู่กับความว่างของดิสค์ และความสามารถของดอสสูงสุด 64,000 บรรทัด จึงทำให้สามารถเขียนโปรแกรมได้ใหญ่กว่าไซค์คิกมาก
จำนวนหน่วยความจำ	192 กิโลไบต์ ถ้าต้องการคอมพายล์โปรแกรมในตัวบรีฟต้องใช้หน่วยความจำอย่างน้อย 320 กิโลไบต์
จำนวนของแมคโครไฟล์	ไม่มีขีดจำกัด ขึ้นอยู่กับหน่วยความจำที่มีอยู่
จำนวนตัวแปร	ไม่มีขีดจำกัด ขึ้นอยู่กับหน่วยความจำที่มีอยู่
จำนวนตัวอักษรต่อบรรทัด	สูงสุด 512 ตัว
การแสดงผล	25, 43 หรือ 50 บรรทัด แล้วแต่การ์ดในการแสดงผลที่ใช้

รูปที่ 1 แสดงขีดความสามารถของบรีฟ

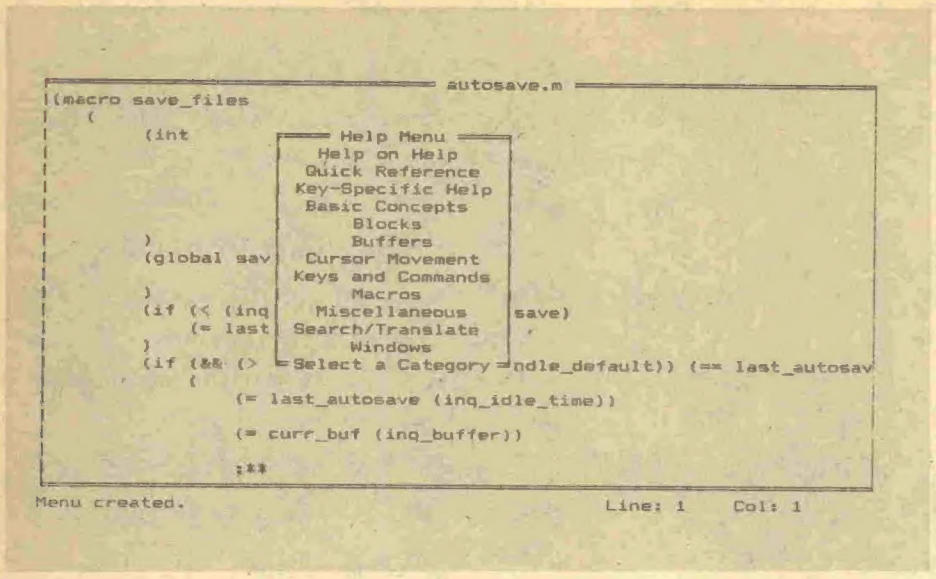
การใช้บรีฟโดยการกดคีย์ ALT เป็นหลัก ในตอนแรกผู้ใช้อาจรู้สึกไม่สะดวกนัก เพราะส่วนมากจะเคยชินกับการใช้เอดิเตอร์ของบอร์แลนด์ แต่เมื่อใช้ไปสักระยะหนึ่ง เราจะรู้จักคุ้นเคยกับคีย์ที่จะใช้ซึ่งคล้ายกับเป็นคำย่อของฟังก์ชันนั้น ๆ ถ้าเราจำคีย์ที่จะกดไม่ได้ เราสามารถขอความช่วยเหลือโดยกด ALT-h (ดูรูปที่ 3 และ 4) หรือดูจาก Quick Reference ในรูปที่ 5 เพื่อความรวดเร็วในการค้นหาคีย์คำสั่งต่าง ๆ ได้

เรามาดูความสามารถพื้นฐาน และความสามารถพิเศษของบรีฟกันดีกว่า ในการเขียน แก้ไข และปรับปรุงโปรแกรมหรือข้อมูลในไฟล์นั้น บรีฟก็ใช้คีย์เหมือนกับเอดิเตอร์ตัวอื่น ๆ เช่น คีย์ลูกศร, DEL, INS, PgUg, PgDown ...ฯลฯ เราสามารถไปที่บรรทัดใด ๆ ก็ได้โดยการกด Alt-g และที่กล่าวไว้ในตอนแรก การกด Alt-U ก็คือฟังก์ชัน Undo เพื่อยกเลิกขั้นตอนการทำงานที่ทำไปแล้วได้ไม่เกิน 30 ขั้นตอน (สามารถเพิ่มได้สูงสุดถึง 300 ขั้นตอน) นอกจากนี้ยังสามารถที่จะแบ่งจอภาพได้เป็นหลาย ๆ หน้าต่าง (WINDOW) โดยใช้กลุ่มคีย์ตามรูปที่ 2

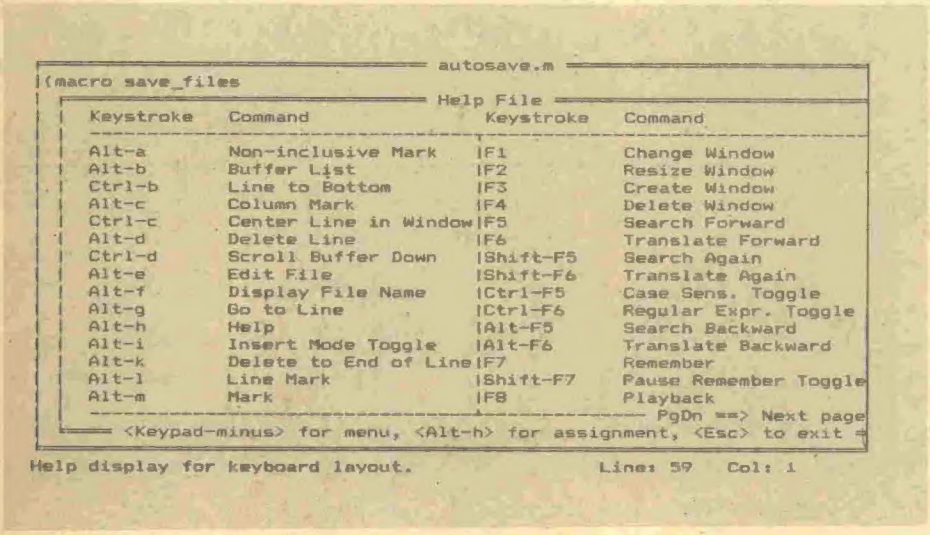
การก๊อบปี้หรือย้ายบล็อก บรีฟใช้รูปแบบของคลิปบอร์ด (clip board) โดยการก๊อบปี้ และย้ายบล็อกผ่าน

รูปที่ 2 แสดงการใช้คีย์เพื่อควบคุมการทำงานของบรีฟ

คีย์	การทำงาน
F1 arrow หรือ shift arrow	เปลี่ยนช่องหน้าต่างที่จะใช้
F2 arrow	เปลี่ยนขนาดของช่องหน้าต่าง
F3 arrow	เปิดช่องหน้าต่าง
F4 arrow	ปิดหน้าต่าง
Home Home	ไปยังตำแหน่งเริ่มต้นของช่องหน้าต่าง
End End	ไปยังตำแหน่งท้ายของช่องหน้าต่าง



รูปที่ 3 แสดงการขอความช่วยเหลือ โดยการกด ALT-h



รูปที่ 4 การขอความช่วยเหลือเกี่ยวกับคำสั่งของคีย์ต่าง ๆ

คลิปบอร์ด การหมายบล็อก (Marked Blocks) ของบรีฟนั้น สามารถทำได้หลายรูปแบบ มีที่ต่างจากเอดิเตอร์ทั่ว ๆ ไป ก็คือ การหมายเป็นคอลัมน์ (Column Mark) ซึ่งบางครั้งก็มีประโยชน์ที่เดียว การค้นหา (Search) และการแทนที่ (Replace) เป็นจุดที่น่าสนใจมาก หรือเรียกได้ว่าเป็นจุดเด่นอย่างหนึ่งของบรีฟเลยก็ว่าได้ ซึ่งมี 2 รูปแบบดังนี้ คือ

1. Case Sensitivity สภาวะ OFF จะเป็นการหาข้อความโดยไม่สนใจตัวใหญ่ตัวเล็ก (เช่นเดียวกับ option U ในเอดิเตอร์ของบอร์แลนด์)

2. Regular expression สภาวะ OFF จะเป็นการหาข้อความตามปกติธรรมดา ส่วนสภาวะ ON จะมีการหาคำได้หลายรูปแบบ เช่น

- การใช้ * และ ? ที่เรียกว่า Wildcard Matching ได้เช่นเดียวกับในดอส เช่น t?p จะหาคำว่า top, tip, ...ซึ่งเครื่องหมาย ? นี้จะแทนด้วยตัวอักษรใดก็ได้ (แต่ไม่ใช่ trap เนื่องจาก ? หนึ่งตัวจะแทนอักษรใด ๆ หนึ่งอักษรเท่านั้น)

t*o จะหาคำว่า too, trowel, sporty orange การหาแบบนี้ คำ ๆ นั้นต้องอยู่บรรทัดเดียวกัน

- การเลื่อนเคอร์เซอร์ คือ \c, การหาการขึ้นบรรทัดใหม่ คือ \n, การหา TAB คือ \t

- การหาคำขึ้นต้นและลงท้ายบรรทัด เช่น

<if คือ หาบรรทัดที่ขึ้นต้นด้วย if

end;> คือ หาบรรทัดที่ลงท้ายด้วย end;

- การหา Matching Set เช่น เมื่อเราเขียนภาษา C ต้องการหา { และ } เราก็ใส่ [{ }]

- การหาแบบ Incremental Search ปกติการหาคำธรรมดาเราต้องใส่คำที่ต้องการหาให้หมดแล้วกด Enter แต่สำหรับการหาแบบนี้จะเป็นการหาแบบทันทีทันใด เช่น เมื่อเรากด P จะหาคำที่มีตัว P อยู่ และเมื่อกด R ต่อ ก็จะหา PR และถ้ากด O ต่ออีกก็จะหา PRO ซึ่งทำได้โดยการกด F10 แล้วพิมพ์ i_search

- การใช้ลักษณะกรุ๊ป เช่น การแทนที่โดย

หาคำ :{[lc]ook}{{out} | {in}}

แทนโดย : \1\0

โดย 0 และ 1 คือ กรุ๊ปที่ 0 และกรุ๊ปที่ 1 กรุ๊ปที่ 0 คือ look หรือ cook กรุ๊ปที่ 1 คือ out หรือ in หรือ [out และ in] ตอนหาคำคือ กรุ๊ป 0 และ 1 เมื่อแทนโดย \1\0 ก็เป็นการสลับที่กันนั่นเอง เช่น lookout เป็น outlook, cookout เป็น outcook, lookin เป็น inlook และ cookin เป็น incook

การหาคำและแทนที่ยังมีรายละเอียด, รูปแบบ และตัวอย่างอีกหลายอย่าง ซึ่งท่านผู้อ่านสามารถหาอ่านได้จากคู่มือการใช้โปรแกรมของบรีฟ

BRIEF Quick Reference Guide

Key Assignments

Basic text commands

Backspace	*Backspace*	Delete Line	*Alt-d*
Insert new line	*Enter*	Delete Character	*Del*
Open Line	*Ctrl-Enter*	Delete to End of Line	*Alt-k*
Back Tab	*Shift-Tab*	Tab Stop	*Tab*

Macros and keystroke recording

Remember	*F7*	Pause Recording	*Shift-F7*
Playback	*F8*	Load Macro File	*F9*
Execute Command	*F10*	Delete Macro File	*Shift-F9*

Manipulating buffers

Buffer List	*Alt-b*	Write Buffer	*Alt-w*
Edit File	*Alt-e*	Read File into Buffer	*Alt-r*
Display File Name	*Alt-f*	Change Output File	*Alt-o*
Previous Buffer	*Alt-minus*	Next Buffer	*Alt-n*
Delete Current Buffer	*Ctrl-minus*		

Windows

Change Window	*F1*	Resize Window	*F2*
Create Window	*F3*	Delete Window	*F4*
	Border Toggle	*Alt-F1*	
	Quick Window Change	*Shift-arrow*	

Cursor movement

Down	↓	Page Down	*PgDn*
Left	←	Page Up	*PgUp*
Right	→	Go to Line	*Alt-g*
Up	↑	Line to Center	*Ctrl-c*
Previous Word	*Ctrl-←*	Next Word	*Ctrl-→*
Line to Top	*Ctrl-t*	Line to Bottom	*Ctrl-b*
Left Side of Window	*Shift-Home*	Scroll Up	*Ctrl-u*
Right Side of Window	*Shift-End*	Scroll Down	*Ctrl-d*
Beginning of Line	*Home*	End of Line	*End*
Top of Window	*Home-Home*	End of Window	*End-End*
	Ctrl-Home		*Ctrl-End*
	Top of Buffer	*Home-Home-Home*	
		Ctrl-PgUp	
	End of Buffer	*End-End-End*	
		Ctrl-PgDn	

Marked blocks

Mark	*Alt-m*	Delete Block	*Del*
Line Mark	*Alt-l*	Print Block	*Alt-p*
Column Mark	*Alt-c*	Write Block	*Alt-w*
Noninclusive Mark	*Alt-a*	Paste from Scrap	*Ins*
Copy to Scrap	*Keypad +*	Cut to Scrap	*Keypad –*
Outdent Block	*Shift-Tab*	Indent Block	*Tab*

Miscellaneous

Escape/Cancel	*Esc*	Help	*Alt-h*
Exit	*Alt-x*	Suspend BRIEF	*Alt-z*
Insert Mode Toggle	*Alt-i*	Display Version ID	*Alt-v*
Next Error	*Ctrl-n*	Previous Error	*Ctrl-p*
Backup File Toggle	*Ctrl-w*	Quote	*Alt-q*
Halt	*Ctrl-Break*	Compile Buffer	*Alt-F10*
Undo	*Keypad**	Repeat	*Ctrl-r*
	Alt-u		

Searching and translating

Search Forward	*F5*	Translate Forward	*F6*
	Alt-s		*Alt-t*
Search Backward	*Alt-F5*	Translate Backward	*Alt-F6*
Search Again	*Shift-F5*	Translate Again	*Shift-F6*
Case Sens. Toggle	*Ctrl-F5*	Regular Exp. Toggle	*Ctrl-F6*

รูปที่ 5 Quick Reference ของบรีฟที่เกี่ยวข้องกับคีย์ต่าง ๆ

การใช้ลักษณะพิเศษของบรีฟ

1. ทำเป็นเวิร์ดโปรเซสซิ่ง (word processing) โดยมีคำสั่งกั้นซ้ายขวา, จัดรูปแบบใหม่ (Reform), จัดกึ่งกลางหน้ากระดาษ ถ้าเราไม่ตั้งระบบใหม่ (default set up) เวิร์ดโปรเซสซิ่งนี้จะใช้กับไฟล์นามสกุล .txt กับ .doc เท่านั้น

2. การจดจำคีย์ที่เรากดไปแล้ว และนำกลับมาใช้โดยการกด F7 เพื่อทำการจำ แล้วกด F7 เช่นกัน เพื่อหยุดการจำ และใช้ F8 เพื่อนำคีย์ที่จำไว้กลับมาใช้

3. การตั้ง TAB โดยการกด F10 (F10 ใช้สำหรับออกคำสั่งกับบรีฟ) เช่น

Tabs 5 8 11

4. การเก็บไฟล์แบบอัตโนมัติ (Auto Save) ซึ่งเป็นลักษณะพิเศษอันที่สำคัญของบรีฟอีกอย่างหนึ่งมีไว้เพื่อป้องกันไฟดับ บรีฟจะทำการเก็บไฟล์ทุก ๆ ช่วงเวลาที่เรากำหนด เช่น 5 นาที ในไฟล์นามสกุล .asv ซึ่งบันทึกการเปลี่ยนแปลงที่เกิดขึ้นก่อนไฟดับ ไฟล์นี้จะถูกลบออกไป เมื่อออกจากบรีฟตามปกติ แล้วเก็บไฟล์ที่กำลังเขียน หรือแก้ไขแทน

5. สามารถตั้งคีย์ใหม่ได้ (Re-configure keyboard) ได้โดยกด F10 แล้วพิมพ์ Keys

6. การใช้ (RUN) โปรแกรมอื่นภายใต้โปรแกรมบรีฟ สามารถใช้คำสั่งดอสได้ภายในตัวของบรีฟเอง โดยการกด F10 แล้วพิมพ์ DOS หน้าข้อความที่ต้องการเช่น

dos dir เป็นการ directory

และ เมื่อต้องการใช้คำสั่งที่มีพารามิเตอร์ให้ใส่เครื่องหมายคำพูดคร่อมคำสั่งนั้น เช่น

dos "type autoexec.bat"

เรามีวิธีใช้โปรแกรมอื่นหรือใช้

คำสั่งของดอส ในขณะที่ใช้บรีฟอยู่ อีกวิธีหนึ่งคือ การกด Alt-Z เพื่อออกสู่ดอสชั่วคราว (Dos shell) และกลับสู่บรีฟโดยพิมพ์ Exit การออกสู่ดอสวิธีนี้ มีข้อพึงระวังอย่างหนึ่งก็คือ เมื่อออกสู่ดอสแล้วห้ามลบไฟล์ที่กำลังทำการเขียนหรือแก้ไข (Edit) หรือแมคโครไฟล์ที่กำลังใช้งานอยู่

7. ใช้ในระบบมัลติทาสกิ้ง (Multitasking System) ได้กับโปรแกรม Microsoft windows, DESQview (Quarterdeck), Double DOS (Softlogic Solution), Task View (Sunny Hill Software)

การเริ่มต้นใช้บรีฟ

บรีฟจะอยู่บนดิสก์ 2 แผ่น ที่ไม่ได้มีการป้องกันการก็อบปี้ เมื่อตั้งระบบ (set up) เรียบร้อยแล้ว เราจะเริ่มใช้บรีฟ ในรูปแบบ

b [file1 file2...] [-flage -flage...]

นั่นคือเราสามารถแก้ไขไฟล์ได้หลาย ๆ ไฟล์ พร้อมกันได้ เช่น

b test.c help.doc screen.h

และชื่อไฟล์นี้ เราสามารถใช้ลักษณะของ wild card ได้เช่น

b *.c *.h ... \ sub \ *.m

หรือ

b ?cc.bat

การใช้บรีฟในการเขียนโปรแกรม

บรีฟใช้การตั้งระบบบนดอสเกี่ยวกับภาษาที่จะเขียน (language package) (ดูรูปที่ 6) ในรูปแบบ

set bpackages = extension, extension...: style, style...; extension, extension...: style...

โดย

extension คือ ชนิดของภาษา เช่น C Pascal Modula ฯลฯ

Style คือ ลักษณะที่จะใช้กับภาษานั้น

สำหรับ style มีดังนี้

1. Regular Indent เป็นลักษณะการย่อหน้าตามบรรทัดก่อนหน้า ซึ่งเป็นลักษณะการย่อหน้าที่เราใช้กันอยู่ในเอดิเตอร์ต่าง ๆ ทำได้โดย

set bpackages = pas, m, mod : r

เป็นการใช้ regular Indent กับภาษาปาสคาล, แมคโครของบรีฟเอง และภาษาโมดูล่า-2

2. Smart Indent เป็นลักษณะการย่อหน้าตามภาษานั้น ๆ เช่น เมื่อเราใช้

set bpackages = c : s

จะได้ผลดังนี้ เมื่อเราพิมพ์

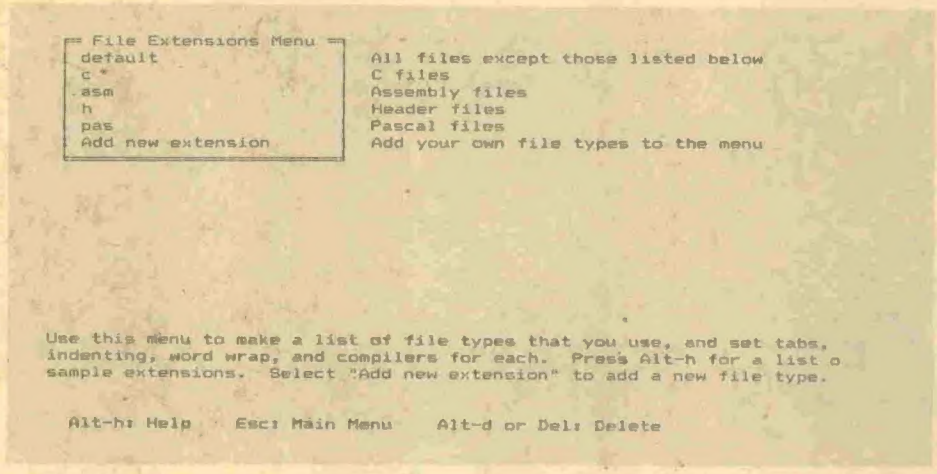
```
if (a = = b)_
```

เมื่อกด Enter

```
if (a = = b)
—
```

เคอร์เซอร์ จะย่อให้ตามลักษณะของภาษาที่เราเขียน เมื่อเขียนต่อเป็น

```
if (a = = b)
  + + b;
—
```



รูปที่ 6 การตั้งระบบ (setup) เราสามารถเลือกภาษาให้ใช้กับบรีฟได้ตามต้องการ

```
echo off
rem Assembler batch file for TASM 1.0
rem Very basic macro to make the compile/correct task easier.  Uses the errorfix
rem macro to locate errors after an unsucessful compile, then attempts to
rem compile again.  Note the invocation of brief with the -e option.
if .%1 == . goto usage
:loop
if .%1 == . goto exit
if not exist %1.asm goto again
if exist %1 goto usage
tasm %1.asm; >%1.err
if errorlevel 1 goto error
del %1.err
goto again
:error
b %1.asm -enext_error
goto loop
:again
shift
goto loop
:usage
echo Usage: c filenames... (no .asm)
:exit
```

รูปที่ 7 โปรแกรมแบตช์ไฟล์สำหรับตัวแปลภาษาแอสเซมบลี

เคอร์เซอร์ จะ outdent ให้อย่างอัตโนมัติ เพราะถือว่าถ้อยแถลงสมบูรณ์แล้ว (statement complete)

3. Templates เป็นลักษณะการใช้คำย่อของภาษานั้น เช่น เมื่อเราพิมพ์ W แล้วกด สเปซบาร์ (space bar) ก็จะได้ while (__)

ซึ่ง Templates สำหรับภาษา C ก็คือ

- if ( )
- else
- else if ( )
- while ( )
- for ( )
- switch ( )
- case:
- return
- do {
  }
  while ( )

ทำได้โดย

set bpackages = c : t

เป็นการทำให้ใช้ภาษา C ในลักษณะ Template แต่คราวนี้ ถ้าเกิดเราต้องการพิมพ์ตัว w ตามด้วย สเปซบาร์โดยไม่ต้องการให้เป็น while ( ) ให้กด Alt-q คั่นระหว่าง w กับ สเปซบาร์

บรีฟสามารถคอมพายล์โปรแกรมที่เราเขียนได้ในตัวเอดิเตอร์ของมันเองโดยไม่ต้องออกจากบรีฟเลย สำหรับตัวคอมพายเลอร์ที่บรีฟมีให้แล้วดูรูปที่ 8

คอมพายเลอร์เหล่านี้บรีฟจะมีแมคโคร (macro file) สนับสนุนการแปลให้แล้ว สำหรับคอมพายเลอร์ที่ท่านผู้อ่านต้องการ แต่บรีฟไม่ได้เตรียมให้ ผู้อ่านสามารถเขียนแมคโครขึ้นสนับสนุนคอมพายเลอร์เหล่านั้นได้ การแปลโปรแกรมภายใต้บรีฟต้องการหน่วยความจำอย่างน้อย 320 กิโลไบต์ ดังนั้นเพื่อป้องกันหน่วยความจำไม่พอ เราควรตั้งแฟล็ก (flag) การใช้หน่วยความจำสำหรับเอดิเตอร์ให้น้อยลง การ

- Brief Macro Compile
- C Compilers : Microsoft, Lattice, Wizard, Computer Innovations, Manx Aztec, Datalight
- Assemblers : Microsoft Macro Assembler
- Pascal Compilers : Oregon Pascal-z, Microsoft
- Cobol Compilers : Realea, Ryon-McFarland. Microsoft, Micro Focus
- dBase Compliers : Nantucket Clipper, Quicksilver, FoxBase
- Logitech
- Microsoft Quick Basic

รูปที่ 8 คอมพายเลอร์ที่บรีฟมีให้แล้ว

```
;---------------------------------------------------------------------------
; Program demo What error when use Brief batch file Assembler
; Use : c getdrive
;---------------------------------------------------------------------------
; Anek sirajariyapiwat
;---------------------------------------------------------------------------
; This routine expects a pointer to be  passed on the stack that points to
; a Turbo Basic integer.  The integer is assigned the value of the current
; logged drive.
;
;  STACK after saving BP
;
;    | 32 bit(segment offset) |
;    | pointer to TB integer  | <----- BP + 6
;    | representing drive id  |
;    |------------------------|
;    | return address that    |
;    | TB will go to after    | <----- BP + 2
;    | completion of routine  |
;    |------------------------|
;    |        saved BP        | <----- BP
;    |                        |
;    |________________________|
;


DosCall  equ  21h       ; equates for module
GetDrive equ  19h


program segment  ; begin program segment
  assume cs:program

  push  bp      ; save bp
  mov   bp, sp
  push  es      ; save es because we'll use it for pointer manipulation

  les   di, [bp + 6h]   ; load pointer to integer into es:di

  mov   ax,7ff0h
  mov   ah, GetDrive
  int   DosCall
  cbw
  mov   es:[di], ax

  pop   es      ; restore registers
  pop   bp
table:
      db   'test Brief batch file
      dw   'You can see everthing when assembler error'.

program ends    ; end program segment

end
```

รูปที่ 9 โปรแกรมแอสเซมบลีสำหรับทดสอบการผิดพลาดที่เกิดขึ้นจากการคอมพายล์โปรแกรม โดยมีแบตช์ไฟล์ช่วย

คอมพายล์โปรแกรมนี้บรีฟสามารถเก็บการผิดพลาด (error) ที่เกิดขึ้นได้ โดยบรีฟจะชี้ที่บรรทัดแรกที่เกิดการผิดพลาด และที่บรรทัดแสดงสถานะ (status line) จะบอกถึงสาเหตุที่เกิด เราใช้ ctrl-n เพื่อดูข้อผิดพลาดตัวต่อไป และ ctrl-p เพื่อดูข้อผิดพลาดทั้งหมด ดังรูปที่ 8 การที่ต้องการหน่วยความจำขนาด 320 กิโลไบต์ บางคนถึงกับร้องว่า น่าเสียดาย จริง ๆ เครื่องมีแค่ 256 กิโลไบต์ อย่าเพิ่งเสียใจครับ บรีฟได้หาทางออกให้ท่านแล้ว โดยบรีฟจะใช้ลักษณะแบตช์-ไฟล์เข้ามาช่วย เพียงแต่ท่านเก็บ (save) โปรแกรมที่เขียนไว้โดยกด ALT-W แล้วออกจากบรีฟ จากนั้นใช้แบตช์ไฟล์ทำการคอมพายล์โปรแกรมของท่าน เมื่อเกิดข้อผิดพลาดจะมีการเรียกบรีฟให้อัตโนมัติ และบรีฟจะแสดงบรรทัดที่เกิดผิดพลาดให้เอง ขณะที่ใช้ ctrl-p เพื่อดูข้อผิดพลาดทั้งหมด เราสามารถใช้เคอร์เซอร์เลื่อนดูข้อผิดพลาด แล้วกด Enter บรีฟจะไปที่บรรทัดที่เกิดข้อผิดพลาดนั้นให้

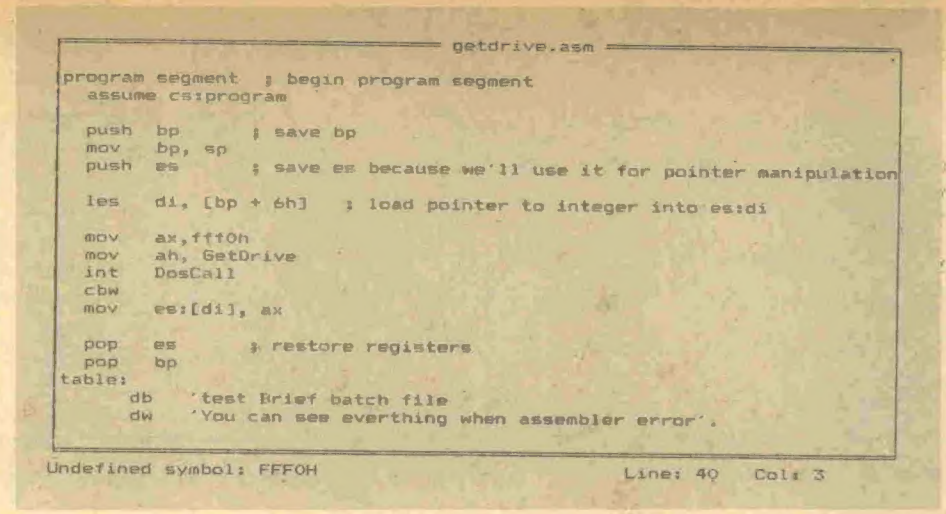
เราลองมาเขียนแมคโครไฟล์สัก 1 ตัวอย่างดู โดยเริ่มจาก

b test.m

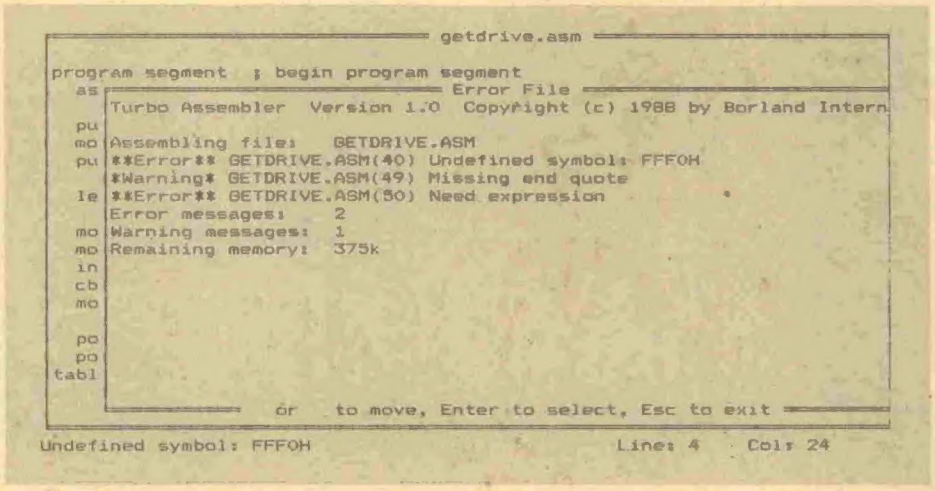
แล้วพิมพ์ ข้อความนี้ลงไป

(macro hello)
(message "Hello, world!")

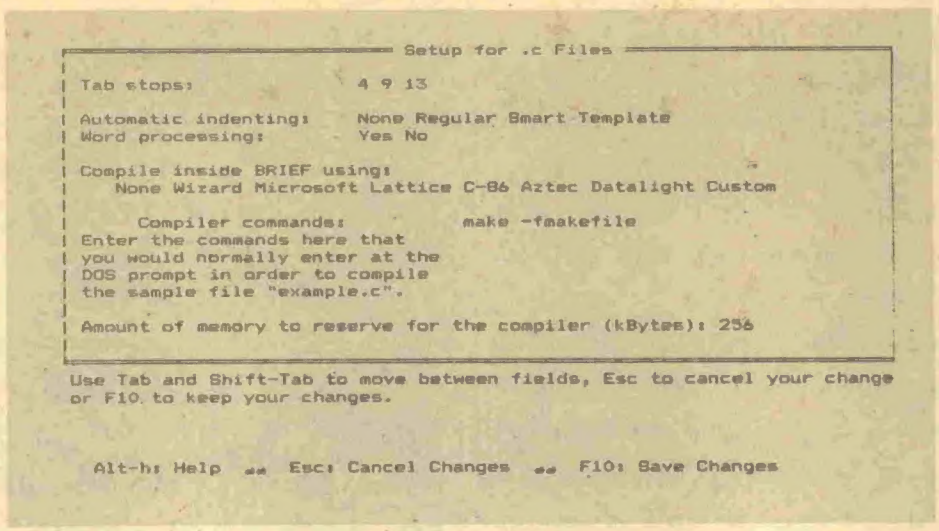
จากนั้นกด Alt-F10 เพื่อทำการแปลแมคโครไฟล์ (ถ้าเป็นภาษาอื่นจะสั่งด้วยแมคโครของภาษานั้น โดยกด F10 แล้วพิมพ์ชื่อแมคโครนั้นลงไป) เมื่อทำการแปลแมคโครเสร็จแล้วก็พร้อมที่จะใช้แมคโครนั้นได้ โดยกด F10 แล้วใส่ชื่อแมคโครคือ hello ท่านจะเห็นข้อความว่า Hello, world! ที่บรรทัดแสดงสถานะแต่ถ้าเป็น



รูปที่ 10 แสดงโปรแกรม getdrive.asm ขณะอยู่ในบรีฟซึ่งคอมพายล์แล้วเกิดการผิดพลาด



รูปที่ 11 การใช้ Ctrl-p เพื่อดูข้อผิดพลาดทั้งหมด



รูปที่12 การใช้คอมพายเลอร์ภาษาซีร่วมกับบรีฟ โดยติดตั้งระบบใหม่ในแบตช์ไฟล์

(macro hello)
(message "Hello, world!")

เมื่อทำการแปลจะเกิดการผิดพลาดขึ้น เคอร์เซอร์จะอยู่ที่บรรทัด message

และที่บรรทัดบอกสถานะจะมีข้อความแสดงข้อมูลเกี่ยวกับบรรทัดนั้น

คราวนี้มาลองดูการทำงานในลักษณะแบตช์ไฟล์กันบ้าง โดยการเขียนแบตช์ไฟล์สำหรับตัวแปลภาษาแอสเซมบลี (TASM เวอร์ชัน 1.0 ของบอร์แลนด์) ดังรูปที่ 7

แบตช์ไฟล์นี้ เขียนในลักษณะที่เมื่อเกิดการผิดพลาดแล้วจะเรียกบรีฟให้อัตโนมัติ เพื่อทำการแก้ไขโปรแกรม เราจะออกจากบรีฟไม่ได้ (เมื่อออกไปแล้ว จะถูกเรียกกลับสู่บรีฟอีก) จนกระทั่งเราแก้ไขโปรแกรมให้ถูกต้องแล้ว หรือแก้ไขแบตช์ไฟล์ใหม่ (สามารถแก้ไขในบรีฟขณะนั้นได้เลยโดยกด Alt-E ตามด้วยชื่อแบตช์ไฟล์) สำหรับท่านผู้อ่านที่ต้องการให้เรียกบรีฟเพียงครั้งเดียว เมื่อจะแก้ไขแล้วจะเกิดการผิดพลาดอีก ก็เพียงแก้ไขบรรทัด ใต้บรรทัด

b %1.ASM -mnext__error

จาก goto loop เป็น goto exit หรือท่านจะเขียนแบตช์ไฟล์ให้สั้นกะทัดรัดและสละสลวยกว่านี้ก็ได้ แต่ที่สำคัญคือ บรรทัด

b %1.ASM -mnext__error

แบตช์ไฟล์อันนี้เวลาใช้ให้พิมพ์ชื่อแบตช์ไฟล์ ตามด้วยชื่อโปรแกรมแอสเซมบลีโดยไม่ต้องใส่นามสกุล .ASM คราวนี้เราลองมาใช้กับโปรแกรมแอสเซมบลี (Getdrive.asm) ดังในรูปที่ ลักษณะโปรแกรมเมื่ออยู่ในบรีฟแสดงดังรูปที่ 10

ใช้ ctrl-n เพื่อหาข้อผิดพลาดตัวต่อไป และเมื่อเรากด ctrl-p จะได้ดังรูปที่ 11 กรณีสุดท้าย คือการใช้คอมพายเลอร์โดยระบุในแบตช์ไฟล์ตั้งแต่เริ่มต้นระบบดังรูปที่ 12 ซึ่งจากในรูปจะเป็นการใช้คอมพายเลอร์ภาษา C ของไมโครซอฟต์

```
                                    autosave.m
(macro save_files
   (
        (int        save_buf
                    curr_buf
                    old_buf
                    saved
                    last_autosave
                    do_autosave
        )
        (global save_buf
                    last_autosave
        )
        (if (< (inq_idle_time) last_autosave)
            (= last_autosave 0)
        )
        (if (&& (> (inq_idle_time) (inq_idle_default)) (== last_autosav
            (
                (= last_autosave (inq_idle_time))

                (= curr_buf (inq_buffer))

                ;**
BRIEF v2.0 - (c) 1984-1987 by UnderWare, Inc.    Line: 56   Col: 1
```

รูปที่ 13 ลักษณะของแมคโครในบรีฟ

```
            PAGE    60,132
            TITLE   SHOW

; Program SHOW.ASM
; Purpose Text file displayer
; Input   File name from command line or prompt
; Output  Display file to screen

            DOSSEG
            .MODEL  small

            INCLUDE dos.inc
            INCLUDE bios.inc

            .STACK  100h

            .DATA

; Status line

            PUBLIC  statline,linenum
statline    DB      " Line:          "
statfile    DB      " File:                    "

Copyright (C) Microsoft Corp 1987, 1988.  All rights reserved
c:\masm\showr.asm (macro) Length=325  Window=(1,1)
```

รูปที่ 14 หน้าตาของเอดิเตอร์ของไมโครซอฟต์ (M.EXE)

```
        PAGE    66,128
        PAGE
;       ********************************************************************
;       *                                                                  *
;       *                    The Norton DOS Editor                         *
;       *                A Command line editor for DOS                     *
;       *                                                                  *
;       *                      By Stanley Reifel                           *
;       *                                                                  *
;       *    Copyright (C) 1986-88  Peter Norton and S. Reifel & Co.       *
;       *                                                                  *
;       *                 01/01/88                    Rev 1.3C             *
;       *                                                                  *
;       ********************************************************************
;
;
MaxSWdt EQU     134                     ;max screen width
MaxLin  EQU     140                     ;max line length to edit
CmdLen  EQU     1000                    ;number of char for previous com
ShrtStr EQU     2                       ;length of strings too short to
;
BS      EQU     08H                     ;ASCII backspace

Line=1      Col=1                A:NDE\NDE.ASM              Insert   WW=0f
```

รูปที่ 15 หน้าตาของนอร์ตันเอดิเตอร์

รูปที่ 16
หน้าตาของ
VEDIT

รูปที่ 17
หน้าตาของ
Slick

Compare Features and Speed

	BRIEF	NORTON Editor	PMATE	VEDIT PLUS
'Off the cuff' macros	No	No	Yes	Yes
Built-in macros	Yes	No	Yes	Yes
Keystroke macros	only 1	No	No	Unlimited
Multiple file editing	20+	2	No	20+
Windows	20+	2	No	20+
Macro execution window	No	No	No	Yes
Pop-up menus	No	No	No	Yes
Execute DOS commands	Yes	Yes	Yes	Yes
Automatic processing of Compiler errors	Yes	No	No	Yes
"Cut and paste" buffers	1	1	1	36
Undo line changes	Yes	No	No	Yes
Paragraph justification	No	No	No	Yes
Convert to/from WordStar	No	No	No	Yes
On-line calculator	No	No	No	Yes
Configurable Keyboard	Hard	No	Hard	Easy
43 line EGA support	Yes	No	No	Yes
Manual size/index	250/No	42/no	469/Yes	380/Yes
Benchmarks in 120K file:				
2000 replacements	1:15 min	34 sec	1:07 min	6 sec
Pattern matching search	20 sec	Cannot	Cannot	2 sec
Pattern matching replace	2:40 min	Cannot	Cannot	11 sec

รูปที่ 18
ตารางเปรียบเทียบ
ความสามารถของ
เอดิเตอร์บางตัว

มาจนถึงจุดนี้ หลายท่านคงบอกว่า ความสามารถของบรีฟนี้ เรียกว่าแทบจะไม่มีที่สิ้นสุด เพราะเราสามารถเขียนแมคโครเพิ่มขึ้นได้ ซึ่งแมคโครนี้เป็นคำสั่งของบรีฟนั้นเอง เช่น แทนที่เราจะกด Alt-h เราลองกด F10 แล้วใส่ Help ซึ่งเป็นแมคโคร เราจะได้การทำงานที่เหมือนกัน แมคโครนี้เขียนเป็นภาษาหนึ่งได้เลย มีลักษณะคล้ายภาษาซี (ดูรูปที่ 13) เพราะฉะนั้นความสามารถของบรีฟจึงขึ้นอยู่กับตัวเราด้วย คือ เราจะเขียนแมคโครได้มากแค่ไหน ดีแค่ไหน และมีประสิทธิภาพเพียงใดนั้นเอง สำหรับเอดิเตอร์ตัวอื่นที่น่าสนใจก็มีอีกหลายตัว เช่น VEDIT PLUS Version 3.0 มีความสามารถเทียบเท่าบรีฟมีลักษณะการใช้เป็นพูลดาวน์เมนู มีป๊อปอัพแอสกีเทเบิ้ล (ASCII table) มีเครื่องคำนวณให้ ซึ่งเป็นตัวที่น่าใช้อีกตัวหนึ่ง ตัวต่อไปคือ M.EXE ซึ่งเป็นเอดิเตอร์ที่ติดมากับแผ่นแอสเซมเบลอร์เวอร์ชัน 5.1 ของไมโครซอฟต์ มีแมคโครเป็นภาษาซี เปิดหน้าต่างได้ 8 หน้าต่าง ต่อไปก็ Slick มีส่วนการค้นหาไฟล์ที่เหมือน FF ของนอร์ตันยูทิลิตี้ อีกตัวก็นอร์ตันเอดิเตอร์ของปีเตอร์นอร์ตัน, PMATE ของ phonix, Epsilon, EMACS, ME และสุดท้ายก็ PI EDIT 2.1 ของ iliad Group (เอดิเตอร์ต่าง ๆ แสดงในรูปที่ 14-18) สาเหตุที่ผมแนะนำบรีฟเพราะคิดว่าความสามารถมันมากมายเพียงพอ เอดิเตอร์รุ่นหลัง ๆ นั้นได้เพิ่มความเร็วในการทำงาน อำนวยประโยชน์แก่ผู้ใช้โดยที่มีการเพิ่มฟังก์ชันมากขึ้น (ดูรูปที่ 18) ซึ่งในบางครั้งก็จำเป็น บางครั้งก็อาจไม่จำเป็น แต่ก็จะเอื้ออำนวยประโยชน์แก่ผู้ใช้ได้มากและหวังว่าท่านผู้อ่านจะมีความสุขกับการกดคีย์และสนุกกับการเขียนโปรแกรมนะครับ สวัสดี